# Introduction to the Quantitative Analysis of Textual Data Using quanteda*

Kenneth Benoit and Paul Nulty

October 2, 2014

## 1 Introduction: The Rationale for quanteda

quantedais an R package designed to simplify the process of quantitative analysis of text from start to finish, making it possible to turn texts into a structured corpus, conver this corpus into a quantitative matrix of features extracted from the texts, and to perform a variety of quanttative analyses on this matrix. The object is inference about the data contained in the texts, whether this means describing characteristics of the texts, inferring quantities of interests about the texts of their authors, or determining the tone or topics contained in the texts. The emphasis of quantedais on *simplicity*: creating a corpus to manage texts and variables attached to these texts in a straightforward way, and providing powerful tools to extract features from this corpus that can be analyzed using quantitative techniques.

The tools for getting texts into a corpus object include:

- loading texts from directories of individual files
- loading texts "manually" by inserting them into a corpus using helper functions
- managing text encodings and conversions from source files into corpus texts
- attaching variables to each text that can be used for grouping, reorganizing a corpus, or simply recording additional information to supplement quantitative analyses with non-textual data
- recording meta-data about the sources and creation details for the corpus.

The tools for working with a corpus include:

- summarizing the corpus in terms of its language units
- reshaping the corpus into smaller units or more aggregated units
- adding to or extracting subsets of a corpus
- resampling texts of the corpus, for example for use in non-parametric bootstrapping of the texts
- Easy extraction and saving, as a new data frame or corpus, key words in context (KWIC)

*This research was supported by the European Research Council grant ERC-2011-StG 283794-QUANTESS. Code contributors to the project include Alex Herzog, William Lowe, and Kohei Watanabe.

For extracting features from a corpus, quantedaprovides the following tools:

- extraction of word types
- extraction of word *n*-grams
- extraction of dictionary entries from user-defined dictionaries
- feature selection through
  - stemming
  - random selection
  - document frequency
  - word frequency
  - and a variety of options for cleaning word types, such as capitalization and rules for handling punctuation.

For analyzing the resulting *document-feature* matrix created when features are abstracted from a corpus, quantedaprovides:

- scaling models, such as the Poisson scaling model or Wordscores
- nonparametric visualization, such as correspondence analysis
- topic models, such as LDA
- classifiers, such as Naive Bayes or *k*-nearest neighbour
- sentiment analysis, using dictionaries

quantedais hardly unique in providing facilities for working with text – the excellent tm package already provides many of the features we have described. quantedais designed to complement those packages, as well to simplify the implementation of the text-to-analysis workflow. quantedacorpus structures are simpler objects than in tm, as are the document-feature matrix objects from quanteda, compared to the sparse matrix implementation found in tm. However, there is no need to choose only one package, since we provide translator functions from one matrix or corpus object to the other in quanteda.

This vignette is designed to introduce you to quantedaas well as provide a tutorial overview of its features.

## 2 Installing quanteda

The code for the quantedapackage currently resides on `http://github/kbenoit/quanteda`. From an Internet-connected computer, you can install the package directly using the devtools package:

```
library(devtools)
if (!require(quanteda)) install_github("quanteda", username="kbenoit")
```

This will download the package from github and install it on your computer. For other branches, for instance if you wish to install the `dev` branch (containing work in progress) rather than the master, you should instead run

```
install_github("quanteda", username="kbenoit", ref="dev")
```

Typically, the `dev` branch of a software package is under active development — so while it contains the latest updates, it is more likely to have bugs. The `master` branch might be missing some of the newer features, but should be more reliable.

# 3 Creating a corpus

## 3.1 Loading Documents into Quanteda

### From a directory of files

The quanteda package provides several functions for loading texts from disk into a quanteda corpus. A very common source of files for creating a corpus will be a set of text files found on a local (or remote) directory. To load in a set of these files, we will load a corpus from a set of text files using information on attributes of the text that have been conveniently stored in the text document's filename (separated by underscores). For example, for our corpus of Irish budget speeches, the filename `2010_BUDGET_03_Joan_Burton_LAB.txt` tells us the year of the speech (2010), the type ("BUDGET"), a serial number (03), the first and last name of the speaker, and a party label ("LAB" for Labour).

To load this into a corpus object, we will use the `corpusFromFilenames` function, supplying a vector of attribute labels that correspond with the elements of the filename.

```
library(quanteda)
data(inaugCorpus)
# inaugCorpus <- subset(inaugCorpus, year=="2010")
```

This creates a new quanteda corpus object where each text has been associated values for its attribute types extracted from the filename:

```
summary(inaugCorpus)
```

```
## Corpus consisting of 57 documents.
##
##             Text Types Tokens Sentences Year  President
##  1789-Washington   595   1430        23 1789 Washington
##  1793-Washington    90    135         4 1793 Washington
##       1797-Adams   794   2318        37 1797      Adams
##   1801-Jefferson   681   1726        41 1801  Jefferson
##   1805-Jefferson   775   2166        45 1805  Jefferson
##     1809-Madison   520   1175        21 1809    Madison
##     1813-Madison   518   1210        33 1813    Madison
##      1817-Monroe   980   3370       122 1817     Monroe
##      1821-Monroe  1192   4459       131 1821     Monroe
##       1825-Adams   962   2915        74 1825      Adams
##     1829-Jackson   500   1128        25 1829    Jackson
##     1833-Jackson   474   1176        29 1833    Jackson
##    1837-VanBuren  1252   3839        95 1837   VanBuren
##    1841-Harrison  1819   8428       215 1841   Harrison
##        1845-Polk  1262   4800       153 1845       Polk
##      1849-Taylor   480   1088        22 1849     Taylor
```

```
##      1853-Pierce  1113   3332       104 1853     Pierce
##    1857-Buchanan   892   2823        89 1857   Buchanan
##     1861-Lincoln  1007   3629       135 1861    Lincoln
##     1865-Lincoln   336    698        26 1865    Lincoln
##       1869-Grant   466   1125        40 1869      Grant
##       1873-Grant   520   1336        43 1873      Grant
##       1877-Hayes   802   2480        59 1877      Hayes
##    1881-Garfield   969   2971       111 1881   Garfield
##   1885-Cleveland   644   1682        44 1885  Cleveland
##    1889-Harrison  1297   4383       157 1889   Harrison
##   1893-Cleveland   798   2013        58 1893  Cleveland
##    1897-McKinley  1181   3960       130 1897   McKinley
##    1901-McKinley   805   2204       100 1901   McKinley
##   1905-Roosevelt   384    984        33 1905  Roosevelt
##        1909-Taft  1374   5427       160 1909       Taft
##      1913-Wilson   627   1699        68 1913     Wilson
##      1917-Wilson   523   1529        59 1917     Wilson
##     1921-Harding  1117   3327       148 1921    Harding
##    1925-Coolidge  1159   4055       196 1925   Coolidge
##      1929-Hoover   997   3558       158 1929     Hoover
##   1933-Roosevelt   708   1880        85 1933  Roosevelt
##   1937-Roosevelt   681   1806        96 1937  Roosevelt
##   1941-Roosevelt   493   1334        68 1941  Roosevelt
##   1945-Roosevelt   257    555        28 1945  Roosevelt
##      1949-Truman   742   2270       118 1949     Truman
##  1953-Eisenhower   846   2444       119 1953 Eisenhower
##  1957-Eisenhower   586   1659        96 1957 Eisenhower
##     1961-Kennedy   534   1363        57 1961    Kennedy
##     1965-Johnson   526   1485        94 1965    Johnson
##       1969-Nixon   708   2122       105 1969      Nixon
##       1973-Nixon   506   1801        72 1973      Nixon
##      1977-Carter   489   1220        52 1977     Carter
##      1981-Reagan   842   2431       135 1981     Reagan
##      1985-Reagan   855   2553       127 1985     Reagan
##        1989-Bush   747   2315       147 1989       Bush
##     1993-Clinton   599   1598        81 1993    Clinton
##     1997-Clinton   716   2157       111 1997    Clinton
##        2001-Bush   584   1581        97 2001       Bush
##        2005-Bush   724   2070       100 2005       Bush
##       2009-Obama   893   2390       119 2009      Obama
##       2013-Obama   773   2092        89 2013      Obama
##
## Source:  /home/paul/Dropbox/code/quanteda/* on x86_64 by paul.
## Created: Fri Sep 12 12:41:17 2014.
## Notes:   .
```

### From a vector of texts

Another method of creating a corpus from texts is to read texts into character vectors, and then create the corpus from these. The

We can also create a labelled corpus using the directory structure in which the files are stored. If the folder names in which the files are stored indicate values for a variable of interest.

**todo corpus from folders***

### 3.2 Structure of a corpus in `quanteda`

A corpus contains attributes and metadata. Metadata is information associated with the entire set of texts, such as the source or date of creation. Metadata can also be used to package supplementary material with a corpus — for example, if the corpus analysis is part of a model that includes other forms of data, they can be included here.

The attributes of a corpus are the texts themselves, and any number of other attributes which may have different values for each text.

## 4 Extracting Features

In order to perform statistical analysis such as document scaling, we must extract a matrix associating values for certain features with each document. In quanteda, we use the dfm function to produce such a matrix. [1].

By far the most common approach is to consider each word type to be a feature, and the number of occurrences of the word type in each document the values. This is easy to see with a concrete example, so lets use the `dfm` command on the full built-in Irish budget speeches corpus. In addition to indexing into the matrix with `:`, you can also view the matrix by clicking on the docMat variable in the RStudio Environment pane, or using the `View()` R command.

[1] dfm stands for document-feature matrix — we say 'feature' as opposed to 'term', since it is possible to use other properties of documents (e.g. ngrams or syntactic dependencies) for further analysis